MACHINE SR_M2
REFINES SR_M1
SEES SR2_Point
VARIABLES

Route_Req
Route_Cel
Route_Occ
Route2Path
Block2Route
Route2OccPath
Point2Pos
Point2Lock

INVARIANTS

inv1: $Point2Pos \in POINT \rightarrow POS$
inv2: $Point2Lock \in POINT \rightarrow ISLOCK$
inv3: $\forall r, po \cdot po \in (Path2Block[\{Route2Path(r)\}] \cap POINT) \wedge r \in dom(Route2Path) \Rightarrow (Point2Pos(po) = Route2Point2Pos(\{r \mapsto po\}))$
inv4: $\forall r, po \cdot po \in (Path2Block[\{Route2Path(r)\}] \cap POINT) \wedge r \in dom(Route2Path) \Rightarrow (Point2Lock(po) = Lock)$

EVENTS

Initialisation ⟨extended⟩
begin
act1: $Route\_Req := \varnothing$
act2: $Route\_Cel := \varnothing$
act3: $Route\_Occ := \varnothing$
act4: $Route2Path := \varnothing$
act5: $Block2Route := \varnothing$
act6: $Route2OccPath := \varnothing$
act7: $Point2Pos := Point2InitPos$
act8: $Point2Lock := POINT \times \{Unlock\}$
end

Event ATS_Request ⟨ordinary⟩ $\widehat{=}$
extends ATS_Request
any
r
where
grd1: $r \notin Route\_Req$
then
act1: $Route\_Req := Route\_Req \cup \{r\}$
end

Event Block_Reserve ⟨ordinary⟩ $\widehat{=}$
extends Block_Reserve
any
r
p
where
grd1: $r \in Route\_Req$
grd2: $r \notin Route\_Cel$
grd3: $r \notin dom(Route2Path)$
grd4: $p \in PATH$
grd5: $p = Route2InitPath(r)$
grd6: $Path2Block[\{p\}] \cap dom(Block2Route) = \varnothing$
then
act1: $Block2Route := Block2Route \cup (Path2Block[\{p\}] \times \{r\})$
end

Event Point_Switch ⟨ordinary⟩ $\widehat{=}$

Before Route_Reserve

any

po

r

where

grd1: $r \notin Route\_Cel$

grd2: $r \notin dom(Route2Path)$

grd3: $po \in Block2Route^{-1}[\{r\}] \cap POINT$

grd4: $Point2Pos(po) \neq Route2Point2Pos(\{r \mapsto po\})$

grd5: $Point2Lock(po) = Unlock$

then

act1: $Point2Pos(po) := Route2Point2Pos(\{r \mapsto po\})$

end

Event Point_Lock ⟨ordinary⟩ $\widehat{=}$

Before Route_Reserve

any

r

po

where

grd1: $r \notin Route\_Cel$

grd2: $r \notin dom(Route2Path)$

grd3: $po \in Block2Route^{-1}[\{r\}] \cap POINT$

grd4: $Point2Pos(po) = Route2Point2Pos(\{r \mapsto po\})$

grd5: $Point2Lock(po) = Unlock$

then

act1: $Point2Lock := \{po \mapsto Lock\} \lhd Point2Lock$

end

Event Route_Reserve ⟨ordinary⟩ $\widehat{=}$

extends Route_Reserve

any

r

where

grd1: $r \in Route\_Req$

grd2: $r \notin Route\_Cel$

grd3: $r \notin dom(Route2Path)$

grd4: $PathConflict[Route2InitPath[\{r\}]] \cap ran(Route2Path) = \varnothing$

grd5: $Block2Route^{-1}[\{r\}] = Path2Block[\{Route2InitPath(r)\}]$

grd6: $\forall po \cdot (po \in (Path2Block[Route2InitPath[\{r\}]] \cap POINT)) \Rightarrow (Point2Pos(po) = Route2Point2Pos(\{r \mapsto po\}))$

grd7: $\forall po \cdot (po \in (Path2Block[Route2InitPath[\{r\}]] \cap POINT)) \Rightarrow (Point2Lock(po) = Lock)$

then

act1: $Route2Path := Route2Path \cup \{r \mapsto Route2InitPath(r)\}$

end

Event Train_Enter ⟨ordinary⟩ $\widehat{=}$

extends Train_Enter

any

r

where

grd1: $r \in dom(Route2Path)$

grd2: $r \notin Route\_Occ$

grd3: $r \notin dom(Route2OccPath)$

then

act1: $Route\_Occ := Route\_Occ \cup \{r\}$

act2: $Route2OccPath := Route2OccPath \cup \{r \mapsto NullPath\}$

end

Event Train_Head_Move ⟨ordinary⟩ $\widehat{=}$

extends Train_Head_Move

any
r
op
b
where
grd1: $r \in Route\_Occ$
grd2: $r \in dom(Route2Path)$
grd4: $b \in Path2Block[\{Route2Path(r)\}]$
grd5: $r \in dom(Route2OccPath)$
grd6: $op = Route2OccPath(r)$
grd7: $b \notin Path2Block[\{op\}]$
then
act1: $Route2OccPath(r) := PathIncrease(op)(b)$
end

Event Train_Rear_Move ⟨ordinary⟩ $\widehat{=}$
extends Train_Rear_Move
any
r
op
b
p
where
grd1: $r \in Route\_Occ$
grd2: $r \in dom(Route2OccPath)$
grd3: $op = Route2OccPath(r)$
grd6: $p = Route2Path(r)$
grd4: $b \in Path2Block[\{op\}] \cap Path2Block[\{p\}]$
grd5: $op \neq NullPath$
grd7: $p \neq NullPath$
then
act1: $Route2OccPath(r) := PathReduce(op)(b)$
act2: $Route2Path(r) := PathReduce(p)(b)$
end

Event Point_Unlock_Release ⟨ordinary⟩ $\widehat{=}$
Before Block_Release
any
r
cp
sp
po
where
grd1: $r \in Route\_Occ$
grd2: $r \in dom(Route2Path)$
grd3: $cp = Route2Path(r)$
grd4: $sp = PathReduce(cp)(po)$
grd5: $r \in dom(Route2OccPath)$
grd7: $po \in POINT$
grd8: $po \notin Path2Block[\{Route2Path(r)\}]$
then
act1: $Point2Lock(po) := Unlock$
end

Event Block_Release ⟨ordinary⟩ $\widehat{=}$
extends Block_Release
any
b
r
where
grd1: $r \in Route\_Occ$
grd2: $b \in Block2Route^{-1}[\{r\}]$

grd3: $b \notin Path2Block[\{Route2OccPath(r)\}]$
grd4: ⟨theorem⟩ $b \notin Path2Block[\{Route2Path(r)\}]$

then

act1: $Block2Route := \{b\} \lhd Block2Route$

end

Event Train_Leave ⟨ordinary⟩ $\widehat{=}$

extends Train_Leave

any

r

where

grd1: $r \in Route\_Occ$
grd2: $r \in dom(Route2Path)$
grd3: $Route2Path(r) = NullPath$
grd4: $Block2Route^{-1}[\{r\}] = \varnothing$

then

act1: $Route\_Occ := Route\_Occ \setminus \{r\}$
act2: $Route2OccPath := \{r\} \lhd Route2OccPath$

end

Event Route_Release ⟨ordinary⟩ $\widehat{=}$

extends Route_Release

any

r

where

grd1: $r \in dom(Route2Path)$
grd2: $Route2Path(r) = NullPath$
grd3: $r \notin Route\_Occ$
grd4: $Block2Route^{-1}[\{r\}] = \varnothing$
grd5: $r \notin dom(Route2OccPath)$

then

act1: $Route2Path := \{r\} \lhd Route2Path$
act2: $Route\_Req := Route\_Req \setminus \{r\}$

end

Event ATS_Cancel ⟨ordinary⟩ $\widehat{=}$

extends ATS_Cancel

any

r

where

grd1: $r \in Route\_Req$

then

act1: $Route\_Cel := Route\_Cel \cup \{r\}$

end

Event Point_Unlock_Cancel ⟨ordinary⟩ $\widehat{=}$

Before Train_Enter

any

r
po

where

grd1: $r \in Route\_Cel$
grd2: $r \notin Route\_Occ$
grd3: $po \in Block2Route^{-1}[\{r\}] \cap POINT$
grd4: $Point2Lock(po) = Lock$

then

act1: $Point2Lock := \{po \mapsto Unlock\} \lhd\!\!\!- Point2Lock$

end

Event Block_Cancel ⟨ordinary⟩ $\widehat{=}$

extends Block_Cancel

any

r

where

grd1: $r \in Route\_Cel$

grd2: $r \in ran(Block2Route)$

grd3: $Path2Block[\{Route2Path(r)\}] \cap dom(Block2Route) = \varnothing$

grd4: $\forall po \cdot po \in Block2Route^{-1}[\{r\}] \cap POINT \Rightarrow Point2Lock(po) = Unlock$

then

act1: $Block2Route := Block2Route \rhd \{r\}$

end

Event Route_Cancel ⟨ordinary⟩ $\widehat{=}$

extends Route_Cancel

any

r

where

grd1: $r \in Route\_Req$

grd2: $r \in Route\_Cel$

grd3: $r \notin Route\_Occ$

grd4: $r \in dom(Route2Path)$

grd5: $Block2Route^{-1}[\{r\}] = \varnothing$

then

act1: $Route\_Req := Route\_Req \setminus \{r\}$

act2: $Route\_Cel := Route\_Cel \setminus \{r\}$

act3: $Route2Path := \{r\} \lhd Route2Path$

end

END